Automatic Quality of Experience Measuring on Video Delivering Networks

Daniel De Vera
InCo, Fac. de Ingeniería
J. Herrera y Reissig 565
11300 Montevideo, Uruguay
daniel.de.vera@gmail.com

Pablo Rodríguez-Bocca
InCo, Fac. de Ingeniería
J. Herrera y Reissig 565
11300 Montevideo, Uruguay
prbocca@fing.edu.uy

Gerardo Rubino
INRIA
Campus de Beaulieu
35042 Rennes, France
rubino@irisa.fr

ABSTRACT

This article describes a full video delivery network monitoring suite. Our monitoring tool offers a new view of a video delivery network, based on the quality as perceived by final users (what is nowadays called Quality of Experience, in short QoE). We measure the perceived quality at the client side by means of the recently proposed PSQA technology, by studying the video flows at the frame level. The developed monitoring suite is a completely free-software application, based on well-known technologies such as Simple Network Management Protocol or Round Robin Databases, which can be executed in various operating systems. In this short article we explain the tool implementation and we present some of the measurements performed with it.

Keywords

QoE, measuring, monitoring, video, VDN

1. INTRODUCTION

There is nowadays an increasing growth of digital Video Delivery Networks (VDN) in different contexts, and especially on the Internet. This is a consequence of the opening of content producers to new business models, the more bandwidth availability on the access network (in the Internet, cellular networks, private IP networks, etc.) and the explosion in development of new hardware capable of reproducing and receiving video streams. For example, it has been observed that the volume of video on the Internet doubles every year, while the demand is increased by a factor of three (see `http://www.researchandmarkets.com`).

A common challenge of any VDN deployment is the necessity of ensuring that the service provides the minimum quality expected by the users. Quality of Experience (QoE) is the name we give today to the overall performance of a system from the users' perspective. QoE is basically a subjective measure of end-to-end performance at the service level, from the point of view of the users. As such, it is also an indicator of how well the system meets its targets [1].

Monitoring tools can be classify into two different categories: active and passive. An active monitoring tool sends traffic through the network for performing its measurements. A passive one uses devices to watch the traffic as it passes through the measuring points. Our platform architecture is an active monitoring tool that use probe nodes distributed in the network, with a centralized data collector based on [3].

The usual approach to assess the performance of a VDN is to analyze a well chosen metric, a metric we know plays an important role in quality. Typical examples are the loss rate of packets, or of frames, delays, etc. In this article we instead address the problem of directly measuring the *perceived* quality by means of the PSQA technology [5, 7, 6]. PSQA is a general procedure that allows the automatic measure of the perceived quality, accurately and, if useful, in real-time.

The article focuses on the implementation of a general platform for monitoring and measuring the quality of experience of a generic Video Delivery Network (including the video quality as perceived by the end users). To reach this objective, a set of open protocols and tools are used. The article is organized as follows. Section 2 introduces the measurement methodology used in our platform. In Section 3, the measurement framework and architecture of the platform is described. In Section 4, the projects using this tool and some of the test already done are shown. The main contributions of this work are then summarized in Section 5.

2. QUALITY MEASUREMENTS

Perceived video quality is, by definition, a subjective concept. The mechanism used for assessing it is called *subjective testing*. It consists of building a panel with real human subjects, which will evaluate a series of short video sequences according to their own personal notion of quality. An alternative is to use a (smaller) panel of experts. In the first case, we will get the quality of the sequences as seen by an average observer. In the second case, we can have a more pessimistic (or optimistic, if useful) evaluation. The output of these tests is typically given as a Mean Opinion Score (MOS). Obviously, these tests are very time-consuming and expensive in manpower, which makes them hard to repeat often. And, of course, they cannot be a part of an automatic process (for example, for analyzing a live streaming in real time, for controlling purposes). There exist standard methods for conducting subjective video quality evaluations, such as the ITU-R BT.500-11 [4].

Some variants included in the standard are: Double Stimulus Impairment Scale (DSIS), Double Stimulus Continuous Quality Scale (DSCQS), Single Stimulus (SS), Single Stimulus Continuous Quality Evaluation (SSCQE), Stimulus Comparison Adjectival Categorical Judgement (SCACJ) and Simultaneous Double Stimulus for Continuous Evaluation (SDSCE).

Other solutions, called *objective tests*, have been proposed. Objective tests are algorithms and formulas that measure, in a certain way, the quality of a stream. The most commonly used objective measures for video are: Peek signal to noise ratio (PSNR), ITS' Video Quality Metric (VQM) [11], EPFL's Moving Picture Quality Metric (MPQM), Color Moving Picture Quality Metric (CMPQM) [8], and Normalization Video Fidelity Metric (NVFM) [9]. With some exceptions, the objective metrics propose different ways of comparing the received sample with the original one, typically by computing a sort of distance between both signals. So, it is not possible to use them in an real-time passive test environment, because the received and the original video are needed at the same time in the same place. Besides, these quality metrics often provide assessments that do not correlate well with human perception, and thus their use as a replacement of subjective tests is limited.

2.0.1 Pseudo Subjective Quality Assessment (PSQA).

We have proposed a hybrid approach between subjective and objective evaluation, which we call Pseudo Subjective Quality Assessment (PSQA) [5]. It is a technique allowing to approximate the value obtained from a subjective test but automatically. The idea is to have several distorted samples evaluated subjectively, that is, by a panel of human observers, and then to use the results of this evaluation to train a specific learning tool (in PSQA the best results come from the Random Neural Networks one [2]) in order to capture the relation between the parameters that cause the distortion and the perceived quality. This method produces good evaluations for a wide range variation of all the quality affecting parameters.

3. THE AUDIT PLATFORM

3.1 An Extensible Measurement Framework

It is possible to distinguish two main components within our measuring system: a set of improved VLC clients and a family of centralized monitoring servers. Each VLC client performs the measuring tasks, and made the measures available to the servers using SNMP or HTTP. In the case of SNMP, a collector server polls the clients to obtain the SNMP MIB values of the measured parameters. In the case of HTTP, the probes reports its measured parameters to the collector server. The advantage with HTTP is that the system can work in an (any) open network environment (SNMP is more suited for a controlled one).

A PSQA server uses these parameters to compute the perceived quality at the clients. A presentation server shows the results to the administrators in a simple way.

The parameters measured in the extended VLC client come from two different types of data sources: either from dynamically calculated information (e.g., video bitrate, I-Frame mean size, P-Frame mean size, B-Frame mean size, codecs detection and so on) or from information included within the own stream.

Normally; all video specifications let the applications to insert their own data into the video flow. In the case of MPEG specifications, it is called "user data". Our measurement framework defines rules about how to tag a stream (for instance, what information should be inserted in the user data, how it should be formatted, and where it should be placed). This inserted information is captured and used by the extended VLC client for the computation f the measures, from which the final quality metric will be computed.

Our framework is extensible, because of its simplicity. This is important for adapting it to new services and networks. The fact that we insert our own information inside the streams gives us flexibility in the definition of the parameters to be reported by the extended VLC clients. For instance, in this way it is possible to send pre-processed source information in the stream, which couldn't be dynamically calculated in the VLC clients. This property is especially useful for us, because the PSQA module can be calibrated by changing, adding or removing some of its input parameters. In particular for the frame losses calculation, we use the stream tagged information. This information should be included as user data in the format defined in Table 1.

Bytes	Field	Value
0 - 4	user_data_start_code	0x000001B2
5 - 6	application_id	0xDDAB
7 - 10	iframes_count	-
11 - 14	pframes_count	-
15 - 18	bframes_count	-

Table 1: User Data Format.

The *user_data_start_code* field is defined in both MPEG-2 and MPEG-4 video standard specifications. The `application_id` field is used in order to identify our application (that is, because other applications could also insert their own user data inside the stream). The `frames_count, pframes_count, bframes_count` counters refer to the I-Frames, P-Frames and B-Frames respectively. The counting process start at the beginning of the ES and they go until the next user data. We send an user data in the header of each GOP (that is, the `user_data` is inserted inside the GOP header). In this way the extended VLC players count the frames received, and when receiving the user data encapsulated information, the latter is extracted and compared with its own measured information (taking them to a common base). By means of this algorithm, frame level losses are detected with a high precision.

An important fact to consider is that the overhead added in the stream by the user data insertion is completely negligible (approximately 19 bytes in 22700 bytes in MPEG-2 and 19 bytes in 281700 bytes in MPEG-4). This makes our solution effective.

3.2 Architecture

Inside the VDN we want to monitor, there are five basic components (Fig. 1): the streaming server, the probes, the data collector server, the PSQA Learning Tool and the Webstat application. The streaming server streams the video's content over the VDN. Probes are VLC players strategically located in the network, taking specific measures and sending reports using SNMP or HTTP. The data collector server polls each probe of the VDN (when using SNMP) in order to gather the probes's reports. The PSQA Module is where the perceptual quality value is computed. Finally `Webstat` provides a web interface for the probes's reports presentation.

3.2.0.1 Streaming Server.

The function of the streaming server is to provide multimedia content to the VDN. This content can be coded using different video specifications (MPEG-2, MPEG-4...), audio

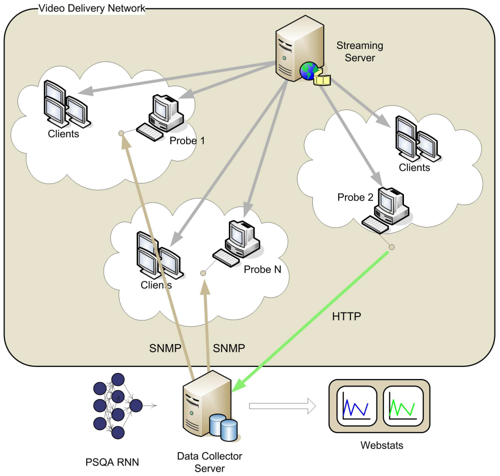


Figure 1: Global architecture – The streaming server, the probes, the data collector server, the PSQA Module (RNN stands for Random Neural Network, the statistical learning tool used) and the Webstat.

specifications (MP3, FLAC...), container formats (MPEG-TS, MPEG-PS, OGM...) and it can be streamed over different transport protocols (HTTP, UDP, RTP, MMS...).

As we mentioned before, the measurement framework requires the user data insertion in the streamed content over the VDN. For this purpose, we have two different alternatives: inserting the user data in the streaming server on the fly, thus using a specific server, or inserting them in a post-encoding process, thus without any need for a specific streaming server.

3.2.0.2 Probes (VLC players).

Probes are VLC players modified in order to measure some specific information. They are strategically located inside the VDN. Basically, a probe is a VLC player with supplementary modules for coordination and data calculation, a SNMP module, an HTTP module and a Logs module. They allow to capture and parse the user-data, to measure and to calculate generic information (like the start and stop of the stream), to offer realtime reports through SNMP or HTTP and to manage a set of rotating logs files with all the relevant probe information.

3.2.0.3 Data Collector Server.

The data collector server is in charge of gathering the probes's information. This application polls each one of the probes in the VDN (with some periodicity) or receives probes reports (when using HTTP module) and saves the data on a Round Robin Database (RRD).

3.2.0.4 PSQA Tool.

This application is in charge of computing the perceptual quality value for each probe in the network. In Fig. 2 we show the mapping between I-frames and P-frames losses into perceived quality. This surface corresponds to a MPEG-4 video stream (in [10], MPEG-2 flows were considered).

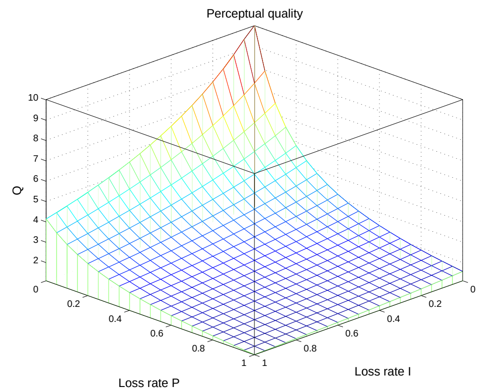


Figure 2: PSQA function (mapping I-frames loss rate and P-frames loss rate into perceived quality).

3.2.0.5 Webstat.

`Webstat` is an application designed to present the data gathered by the data collector server to administrators and managers of the VDN. It offers reports at different levels and types of views. It is possible to generate reports focused on a particular stream or on a specific client (where there could be more than one active stream), or perhaps on the entire network (where there could be more than one client). Finally, it is possible to display the results at the frame level, possibly per frame type (I, P, B), or at the PSQA level.

In Fig. 3, we present an example of the results shown by the `Webstat` application.

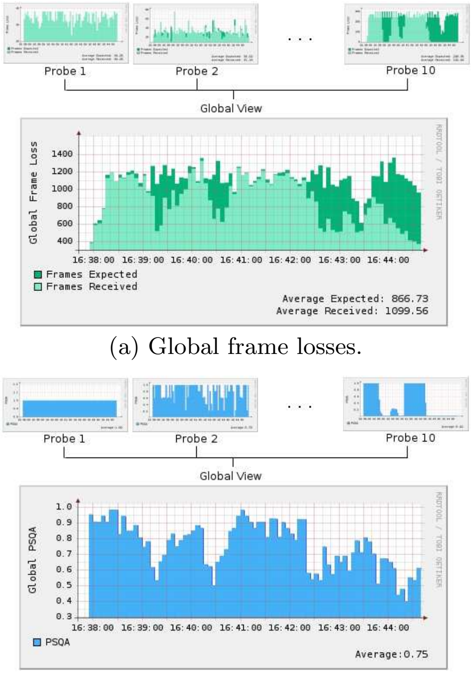
3.2.0.6 Implementation.

As mentioned before, both the probes and the streaming server are based on a VLC Player. The `libavcodec (ffmpeg)` library was used in order to work with MPEG-2 and MPEG-4 video specifications. As container format we worked with MPEG-TS using the functionalities provided by `libdvbps`. We streamed over HTTP and UDP using VLC internal modules. In order to report the probes's measures through SNMP we used the Net-SNMP library. We used the RRDtool to generate the statistical graphs. The data collector server was written in PHP and we used the following PHP extensions: `php4-snmp` and `php4-rrdtool`. `Webstat` is also written in PHP; it uses MySQL as relational database and it runs over Apache. All libraries and applications mentioned above are free-software and they can be executed in Microsoft Windows, Unix, Linux, etc.

4. EVALUATION (PROJECTS USING THIS TOOL)

This tool has been used in the Gol!P2P project (`http://p2ptv.gforge.inria.fr`) and is currently used in the Goalbit project (`http://goalbit.sourceforge.net/`). Both peer to peer video streaming networks focus on broadcast TV (delivery of live content).

It has been tested in different contexts, over a local area network(LAN) injecting looses at the network layer and at the server level (dropping frames in the streaming server), also in an real context over Planet-lab. For more information see [10].



(a) Global frame losses.

(b) Global perceptual quality (QoE).

Figure 3: Typical global view of the entire Video Delivery Network.

5. CONCLUSION

This article presents an effective monitoring and measuring tool that can be used by VDN managers and administrators to assess the streaming quality inside the network. With this tool it is possible to automatically monitor different sets of parameters of the streams, in real-time if necessary, including the perceived quality as seen by the final users, thanks to our version of the recently proposed PSQA technology.

Another important feature of our tool is that it is not dependent on the considered VDN. It was designed as a generic implementation, in order to be able to use it with multiples VDN architectures. Moreover, it can be associated with most common management systems since it is built over the SNMP standard. Alternatively, it can rely on HTTP to make it easy to use in an open network environment. Another feature is that the induced overhead is negligible. Finally, the tool is a free-software application that can be executed on several operating systems.

6. REFERENCES

[1] DSL Forum Technical Work WT-126. Video services quality of experience (qoe) requirements and mechansims, Apr. 2007.

[2] E. Gelenbe. Random Neural Networks with Negative and Positive Signals and Product Form Solution. *Neural Computation*, 1(4):502–511, 1989.

[3] IETF Network Working Group. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks (RFC 3411), Dec. 2002.

[4] ITU-R Recommendation BT.500-11. Methodology for the subjective assessment of the quality of television pictures, June 2002.

[5] S. Mohamed and G. Rubino. A Study of Real–time Packet Video Quality Using Random Neural Networks. *IEEE Transactions On Circuits and Systems for Video Technology*, 12(12):1071 –1083, Dec. 2002.

[6] S. Mohamed, G. Rubino, and M. Varela. A method for quantitative evaluation of audio quality over packet networks and its comparison with existing techniques. In *Proceedings of the Measurement of Speech and Audio Quality in Networks workshop (MESAQIN'04)*, Prague, Czech Republic, June 2004.

[7] G. Rubino. Quantifying the Quality of Audio and Video Transmissions over the Internet: the PSQA Approach. In *Design and Operations of Communication Networks: A Review of Wired and Wireless Modelling and Management Challenges*, Edited by J. Barria. Imperial College Press, 2005.

[8] C. van den Branden Lambrecht. Color Moving Picture Quality Metric. In *Proceedings of the IEEE International Conference on Image Processing*, Sept. 1996.

[9] C. van den Branden Lambrecht. *Perceptual Models and Architectures for Video Coding Applications*. PhD thesis, EPFL, Lausanne, Swiss, 1996.

[10] D. D. Vera, P. Rodríguez-Bocca, and G. Rubino. QoE Monitoring Platform for Video Delivery Networks. In *7th IEEE International Workshop on IP Operations and Management (IPOM'07)*, San José, California, United States, October 31 - November 2 2007.

[11] S. Voran. The Development of Objective Video Quality Measures that Emulate Human Perception. In *IEEE GLOBECOM*, pages 1776–1781, 1991.